REMARKS

Claims 1-7, 9-11, 13, 15-22 and 24-35 were examined and reported in the Office Action. Claims 1-7, 9-11, 13, 15-22 and 24-35 are rejected. Claims 1-7, 9-11, 13, 15-22 and 24-35 remain.

Applicant requests reconsideration of the application in view of the following remarks.

I. 35 U.S.C. §103

A. It is asserted in the Office Action that claims 1-7, 9-10 and 29-35 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy et al. ("Slipstream Processors: Improving both Performance and Fault Tolerance", ASPLOS, pp. 257-268, Nov. 2000) ("Sundaramoorthy") in view of U. S. Patent No. 6,757,811 issued to Mukherjee ("Mukherjee") in view of Hennessy and Patterson ("Computer Architecture A Quantitative Approach", Morgan Kaufmann, 1996) ("Hennessy"). Applicant respectfully traverses the aforementioned rejections for the following reasons.

According to MPEP §2142

> [t]o establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. (In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Further, according to MPEP §2143.03, "[t]o establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. (In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)." *"All words in a claim must be* *considered* in judging the patentability of that claim against the prior art." (In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970), emphasis added.)

Applicant's claim 1 contains the limitations of

> [a]n apparatus comprising: a first processor and a second processor each having a scoreboard and a decoder; a plurality of memory devices coupled to the first processor and the second processor; a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer; a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and a plurality of memory instruction buffers coupled to the first processor and the second processor; wherein the first processor and the second processor perform single threaded applications using multithreading resources, and the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant's claim 29 contains the limitations of

> [a] system comprising: a first processor and a second processor each having a scoreboard and a decoder; a bus coupled to the first processor and the second processor; a main memory coupled to the bus; a plurality of local memory devices coupled to the first processor and the second processor; a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer; a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and a plurality of memory instruction buffers coupled to the first processor and the second processor, wherein the first processor and the second processor perform single threaded applications using multithreading resources, the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is

> not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Sundaramoorthy discloses a multiprocessor system that executes two (i.e., multiple streams/threads) pseudo-redundant programs on separate processors on the same chip. The two redundant programs have a different amount of instructions. (Sundaramoorthy, column 2, lines 34-56). That is, one of the programs has more instructions than the other. (Sundaramoorthy, page 258, first column, lines 40-55). Sundaramoorthy further discloses that the A-stream has fewer instructions than the R-stream, which receives information from the A-stream. Both programs run in parallel on two processors. Sundaramoorthy, however, does not teach, disclose or suggest "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

In other words, Sundaramoorthy executes multiple-streams on two separate processors where each stream has a different number of instructions, as opposed to Applicant's claimed invention, which executes a single thread application without converting or duplicating the single thread application to a multiple-thread application when using multiple-thread processing resources, where the single thread application is executed on two processors without changing the number of instructions (i.e., an exact duplicate).

Hennessy discloses using scoreboarding to aid in allowing instructions to execute out of order. Sundaramoorthy, however, is directed to finding instructions that do not effect final program output and removes these instructions from a second stream, for example redundant instructions. Therefore, the combination of the two prior art documents would not result in Applicant's claimed invention. Further, Hennessy does not teach, disclose or suggest "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

Mukherjee is relied upon for disclosing a multi-threaded processor executes two single threads, where one thread executes slightly ahead of the other for fault tolerance. The two threads are duplicates of one another. Mukherjee deals with multiple threads in a multi-threading processor, not single threaded processes. Mukherjee further asserts that the leading and trailing thread are executed on the same processor. Mukherjee does not teach, disclose or suggest

> the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Moreover, it is asserted in the Office Action that it would benefit Sundaramoorthy to run two streams having the same amount of instructions with one running ahead of the other. This completely opposes the disclosure of Sundaramoorthy as one of ordinary skill in the art would know that streams of the same amount of instructions only adds latency.

Therefore, even if Sundaramoorthy were combined with Mukherjee and Hennessy, the resulting invention would still not include all of Applicant's claimed limitations. And, therefore, there would be no motivation to combine Sundaramoorthy, Mukherjee with Hennessy.

Moreover, by viewing the disclosures of Sundaramoorthy, Mukherjee and Hennessy, one can not jump to the conclusion of obviousness without impermissible hindsight. According to MPEP 2142,

> [t]o reach a proper determination under 35 U.S.C. 103, the examiner must step backward in time and into the shoes worn by the hypothetical 'person of ordinary skill in the art' when the invention was unknown and just before it was made. In view of all factual information, the examiner must then make a determination whether the claimed invention 'as a whole' would have been obvious at that time to that person. Knowledge of applicant's disclosure must be put aside in reaching this determination, yet kept in mind in order to determine the 'differences,' conduct the search and evaluate the 'subject matter as a whole' of the invention. The tendency to resort to 'hindsight' based upon applicant's disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art.

Applicant submits that without first reviewing Applicant's disclosure, no thought, whatsoever, would have been made to the limitations of

> the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Neither Sundaramoorthy, Mukherjee, Hennessy, and therefore, nor the combination of the three, teach, disclose or suggest the limitations contained in Applicant's amended claims 1, 20 and 29, as listed above. Since neither Sundaramoorthy, Mukherjee, Hennessy, nor the combination of the three teach, disclose or suggest all the limitations of Applicant's amended claims 1 and 29, claims 1 and 29 are not obvious over Sundaramoorthy in view of Mukherjee and Hennessy since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly or indirectly depend from amended claims 1 and 29, namely claims 2-7 and 9-10, and 30-35, respectively, would also not be obvious over Sundaramoorthy in view of Mukherjee and Hennessy for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 1-7, 9-10 and 29-35 are respectfully requested.

B. It is asserted in the Office Action that claims 11, 13, and 15-19 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy and in further view of Akkary (WO 99/31594). Applicant respectfully traverses the aforementioned rejection for the following reasons.

Applicant's claim 11 contains the limitations of

> [a] method comprising: executing a plurality of instructions in a single thread by a first processor; executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor, transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information; transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction

> has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant's claims 13 and 15-19 directly depend on amended claim 11.

Applicant's claim 11 contains similar limitations as claims 1, 20 and 29 above. Namely,

> a single thread by a first processor; executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor, transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information; transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Neither Sundaramoorthy, Mukherjee, Hennessy, and therefore, nor the combination of the three teach, disclose or suggest the limitations contained in Applicant's amended claim 11.

Akkary discloses a system for ordering loads and stores in a multi-threaded processor using load and store buffers. Applicant is well aware of Akkary as Akkary is owned by Applicant's Assignee. Akkary does not teach, disclose or suggest

> the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Therefore, combining Akkary with Sundaramoorthy, Mukherjee and Hennessy would still not result in Applicant's claim 11 as the combination would still run single threaded applications without converting to explicit multi-threaded applications, where the two single threaded processes are executed on different processors and the two single-threaded processes have the same amount of instructions and use multithreading resources.

Neither Sundaramoorthy, Hennessy, Mukherjee, Akkary, and therefore, nor the combination of the four, teach, disclose or suggest the limitations contained in Applicant's claim 11, as listed above. Since neither Sundaramoorthy, Hennessy, Mukherjee, Akkary, nor the combination of the four, teach, disclose or suggest all the limitations of Applicant's claim 11, Applicant's claim 11 is not obvious over Sundaramoorthy in view of Mukherjee, Hennessy, and further in view of Akkary since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly depend from claim 11, namely claims 13 and 15-19, would also not be obvious over Sundaramoorthy in view of Mukherjee, Hennessy and further in view of Akkary for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 11, 13, and 15-19 are respectfully requested.

C. It is asserted in the Office Action that claims 20-22, and 24-28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy in view of Akkary, as applied above, and in further view of Tanenbaum "Structured Computer Organization," Prentice-Hall, 1984, pp. 10-12 ("Tanenbaum"). Applicant respectfully traverses the aforementioned rejections for the following reasons.

Applicant's claim 20 contains the limitations of

> [a]n apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising: executing a single thread from a first processor; executing said single thread from a second processor as directed by the first processor, the second processor executing instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a first buffer, and written by said second processor, said tracking executed by said second processor, the first buffer being a register file buffer, and clearing a store validity bit and setting a mispredicted bit in a load entry in a second buffer if a replayed store instruction has a matching store identification (ID) portion, the second buffer being a trace buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant has similar limitations above in section I(B) regarding Sundaramoorthy in view of Mukherjee, Hennessy and Akkary.

Tanenbaum is relied on for asserting that "any instruction executed by hardware can also be executed in software."

> the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Therefore, combining Tanenbaum with Akkary, Sundaramoorthy, Mukherjee and Hennessy would still not result in Applicant's claim 20 as the combination would still run single threaded applications without converting to explicit multi-threaded applications, where the two single threaded processes are executed on different processors and the two single-threaded processes have the same amount of instructions and use multithreading resources.

Neither Sundaramoorthy, Mukherjee, Hennessy, Akkary, Tanenbaum, and therefore, nor the combination of the five, teach, disclose or suggest the limitations contained in Applicant's claim 20, as listed above. Since neither Sundaramoorthy, Mukherjee , Hennessy, Akkary, Tanenbaum, nor the combination of the five, teach, disclose or suggest all the limitations of Applicant's claim 20, Applicant's claim 20 is not obvious over Sundaramoorthy in view of Mukherjee, Hennessy and Akkary and further in view of Tanenbaum since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly or indirectly depend from claim 20, namely claims 21-22 and 24-28, would also not be obvious over Sundaramoorthy in view of Mukherjee, Hennessy and Akkary, and further in view of Tanenbaum for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 20-22, and 24-28 is respectfully requested.

CONCLUSION

In view of the foregoing, it is believed that all claims now pending, namely 1-7, 9-11, 13, 15-22 and 24-35, patentably define the subject invention over the prior art of record and are in condition for allowance and such action is earnestly solicited at the earliest possible date.

If necessary, the Commissioner is hereby authorized in this, concurrent and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2666 for any additional fees required under 37 C.F.R. §§ 1.16 or 1.17, particularly extension of time fees.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR, & ZAFMAN LLP

Dated: December 21, 2005

Steven Laut, Reg. No. 47,736

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail with sufficient postage in an envelope addressed to: Mail Stop AF, Commissioner for Patents, P. O. Box 1450, Alexandria, Virginia 22313-1450 on December 21, 2005.

Jean Svoboda